smxNS ™

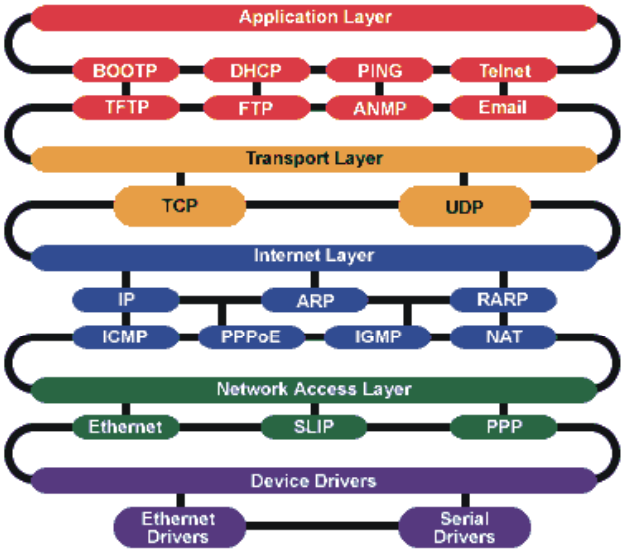
smxNS is a compact TCP/IP networking stack for use with real-time, embedded applications. It provides drop-in support for many processors and tool chains. A large selection of networking protocols is available for use with smxNS. Complete source code and test programs are provided. smxNS is licensed per application and is royalty free.

FEATURES

- Embedded TCP/IP
- Optimized for use with the SMX® Real Time Operating System
- Processor independent
- User configurable
- Royalty-free
- Small footprint (25K - 49K bytes code space depending upon the processor and compiler)
- Ethernet drivers for latest SoCs
- ROMable and reentrant
- Includes complete source code in ANSI C
- Berkeley sockets API or
- Dynamic Protocol Interface (DPI)
- New: mDNS, IPAlias, VLAN

Components

smxNS comes with complete source code that covers all layers in a TCP/IP network stack, from high level network protocols to device drivers.



Application Layer

c = client, s = server, * = included in base package

BOOTP c*	Allows a client to obtain its IP address and boot file from a BOOTP server
DHCP c*/s	Dynamic Host Control Protocol. Allows a client to obtain a temporary IP address.
Ping c*	Uses ICMP to report if a remote host is responding.
Telnet s*	Acts as a command processor.
TFTP c*/s*	Trivial File Transfer Protocol.
FTP c/s	File Transfer Protocol.
mDNS	Multicast DNS Responder
POP c	Receive email from an email server.
SMTP c/s	Send email (c) or accept email (s).
SNMP	Simple Network Management Protocol agent v1, v2, and v3.

Transport Layer

TCP*	Transmission Control Protocol – for reliable connection. Required for FTP and Telnet.
UDP*	User Datagram Protocol – unreliable (best effort), connectionless transfer. Used by DHCP, DNS, SNMP, and TFTP.

Internet Layer

IP*	Internet Protocol v4. Multiple IP addresses can be associated with a single network interface (IP Alias).
ARP c*/s*	Address Resolution Protocol – Converts IP to Ethernet address.
RARP c*/s*	Reverse Address Resolution Protocol – Converts Ethernet to IP address.
ICMP*	Internet Connection Management Protocol
IGMP	Internet Group Management Protocol (used for multicast).
PPPoE c/s	PPP over Ethernet. Allows connecting to an ISP via Ethernet, when available.
NAT	Network Address Translation – allows sharing an IP address among multiple devices on a private network.

Network Access (Link) Layer

Ethernet*	Interface to Ethernet drivers, which process the Ethernet protocol. Support is included for sending and receiving 802.1Q VLAN tags in Ethernet headers.
SLIP*	Serial Line Internet Protocol. For connection via serial lines.
PPP	Point to Point Protocol. For connection to ISPs (Internet Service Providers). Includes CHAP and MS-CHAP authentication protocols.

Device Drivers

Ethernet	Popular Ethernet controllers are supported.
Serial	UART drivers are provided.

Advantages

Flexible Configuration

Because code and data space are at a premium in an embedded system, smxNS can be configured to use only those clients, servers, protocols, devices drivers, and link layers needed by an application. By judiciously selecting features and capabilities, the smxNS TCP/IP stack can be reduced to as little as 25KB code space, depending upon the processor.

Clean Modular Design

smxNS implements a clean, modular design. Network layers provide standardized APIs that present a common set of entry points for each layer in the network from the transport layer down to the device driver.

Choice of API

Network applications can be written to use either the standard Berkeley Sockets API, or our proprietary Dynamic Protocol Interface (DPI). DPI is a simplified API that takes little code space and is easy to use. Sample applications demonstrating both of these APIs are included in smxNS. The Sockets API facilitates interfacing with third party or legacy software. Both APIs can be used simultaneously.

Zero-Copy

smxNS supports zero-copy packet processing for optimum performance and efficiency. Information to be sent is assembled in a single buffer, which is passed from layer to layer with no copying, and then sent directly by the network controller. Likewise, each received packet is loaded into a buffer by the network controller, which is then read by each layer and delivered to the application with no copying. If the network controller does not support DMA, there is one copy operation, from the buffer in the network controller to the smxNS buffer in memory. Use of the zero-copy interface is demonstrated in example code that ships with smxNS.

Routing

smxNS supports up to 256 network interfaces, and can route packets between interfaces. A network interface is defined as a physical connection to a network (e.g. Ethernet) or to a serial link. Hence, smxNS could route packets from an Internet Service Provider (ISP) to devices connected to an Ethernet Local Area Network (LAN). ARP would be used to convert between IP and Ethernet addresses.

Built in Tests

smxNS comes with a set of confidence tests and debugging aids that permit verification of proper operation on any platform. Several state display functions allow easily checking the operation of features implemented in an application and tracking down unexpected behavior.

Ethernet Drivers

A choice of Ethernet drivers and link layers is available for smxNS. These drivers provide support for the most popular network controllers including Fast Ethernet (100 Mbit/sec) controllers, giving an "out-of-the-box" solution to your networking needs.

- AMD AM7990 / 79C96x
- Atmel AT91SAM7, SAM9, RM9200
- Cirrus Logic EP93xx
- ColdFire FEC, 548x/7x FEC, and MAC-NET
- Crystal CS8900
- Davicom DM9000
- DEC DC21140
- Intel i82557
- Kinetis MAC-NET
- Novell NE2000
- NXP LPC1/2/3xxx EMAC
- NXP LPC18xx EMAC
- Realtek RTL8139
- SMSC LAN91C111
- STMicro STM32F107 / 2x7 / 4x7
- STMicro STR9
- TI AM1xxx, AM35xx
- TI Stellaris LM3S

New drivers are being added steadily, so call if you do not see the driver you need.

Cryptographic Functions

smxNS includes cryptographic functions to support authentication and encryption in add-on components as follows:

- PPP: LAN Manager compatible challenge response, MD4 and MD5 for authentication
- SNMPv3 Agent: MD5 and SHA for authentication, DES for encryption
- Web Server: MD5 for authentication

Customers outside the United States should contact Micro Digital to discuss the current arrangements regarding export licensing for these cryptographic functions.

Other Network Protocols

A rich selection of other network protocols is available for use with smxNS. The following is a brief discussion of each.

DHCP Client

smxNS includes a DHCP client so that a system can be deployed on a network without needing to hand-configure an IP address. If the system fails to find a DHCP server, it can continue trying, fall back to a fixed IP address, or use a link-local IP address. smxNS includes Address Conflict Detection (RFC 5227) to ensure that the address that is adopted does not conflict with other systems on the local network.

DHCP Server

smxNS DHCP (Dynamic Host Control Protocol) server delivers host configuration parameters, such as IP address, to a client host.

HTTP Client

smxNS HTTP (Hypertext Transfer Protocol) client allows retrieving a web page from a web server. This is not intended for use as a general purpose browser, but rather as a mechanism for automated retrieval of information that is available via a web page, for configuration and control of a remote device.

IGMP

smxNS IGMP (Internet Group Management Protocol) allows sending messages to multiple hosts in a multicast group. This is more efficient than broadcasting messages to all hosts in a network.

mDNS Responder

Multicast DNS is a protocol that allows a host on the network to discover services provided by other hosts. For example, a desktop computer could find a printer on the network without requiring the user to enter the address of the printer. This allows you to create products that simply plug in to a network to add services. When used together, the mDNS Responder and the DHCP client will obtain an IP address and advertise a service at that address.

NAT

smxNS NAT (Network Address Translation), which is written based on industry standard RFC 2663, enables local-area network (LAN) hosts to use one set of IP addresses for internal traffic and communicate with external networks by presenting a single routable IP address.

- It allows several local devices to share a single external IP address and combine multiple connections into a single Internet connection.
- It provides firewall protection by hiding internal IP addresses from the Internet
- There is no need to reconfigure the NAT-enabled router when new devices are added to the LAN.

POP3

smxNS POP3 (Post Office Protocol) client enables an embedded device to receive email from an email server. It supports MIME attachments.

PPP

smxNS PPP (Point to Point Protocol) is based on industry standard RFC 1661 for establishing a link to a single remote host, such as an Internet Service Provider (ISP), often via a serial or modem link. PAP & CHAP are supported for authentication. A dialer and sample dial-up scripts are included, as well as a runtime script interpreter that allows the system to be tailored to proprietary modems or ISP configurations. The system can be localized by pulling in certain script parameters (such as a dial up number) from non-volatile memory.

PPPoE

smxNS PPPoE (PPP over Ethernet) is based on industry standard RFC 2516 for connecting the users on an Ethernet network to the Internet through a shared access to a broadband medium, such as a single DSL line, wireless device, or cable modem. smxNS PPPoE can be configured to support operation as either a PPPoE Host or a PPPoE Access Concentrator. This means that smxNS can be incorporated into a device that accesses the Internet through a PPPoE router, or it can be incorporated into a PPEoE router, acting as an Access Concentrator that serves PPPoE hosts on a local Ethernet network

The smxNS PPPoE implementation requires virtually no more knowledge on the part of the end user than that required for standard dial-up Internet access. In addition, PPPoE requires no major changes in the operational model for Internet Server Providers (ISPs) and carriers.

SMTP

smxNS SMTP (Simple Mail Transport Protocol) client enables an embedded device to send email. It supports MIME attachments. SMTP server is also available to accept and forward email from SMTP clients.

SNMP Agent

smxNS SNMP (Simple Network Management Protocol) Agent allows a device to be accessed and controlled from a central SNMP network monitor. (Network monitoring software is provided by other companies.) smxNS SNMP Agent includes a MIB compiler and may be used with both MIB-II and user-created MIBs. The MIB compiler converts user MIBs to the internal format needed by the SNMP agent. Options offered are SNMP v1/2 and SNMP v1/2/3. SNMPv1 has no security; SNMPv2 has moderate security; and SNMPv3 has high security.

Web Server

The smxNS Web Server allows an embedded system to present real-time information to desktop or mobile web browsers. It is a full featured HTTP server that is HTTP 1.0 compliant and includes HTTP 1.1 features. smxNS provides a number of features that go beyond basic serving of static pages.

Dynamic Content

Server Side Includes (SSI)
Server Side Includes are a mechanism that allows the web server to dynamically insert content into a web page based on directives that are placed inside special HTML comments. Directives exist for pulling in canned blocks of text, displaying a variable with flexible formatting, or generation of arbitrary text using a user defined function.

Common Gateway Interface (CGI)
The Common Gateway Interface is a convention for interfacing a web server with a stand-alone program that runs on the same system. The stand-alone program may receive parameters associated with a web browser

request and generate information that is piped back through the web server for delivery to the browser.

This convention has its roots in UNIX systems. The smxNS Web Server makes a few adaptations to fit this model to an embedded environment. Instead of a stand-alone program, an smxNS CGI function is linked directly in to the system image. Traditional CGI programs receive input through environment variables or stdin and return information to the server through stdout. With smxNS CGI functions, the input from a GET request is retrieved with the Ngetenv( ) function, and the input from a POST request is retrieved through the Nmakeword( ) function. SmxNS CGI functions send their output through the rplyprintf( ) function. The resulting smxNS CGI function ends up looking very similar to a UNIX CGI program written in C.

CGI programs are often used to process HTML forms, where the fields in the form are provided as inputs, and the output from the CGI program becomes the page that is displayed when the user submits the form. In an embedded system, since control is passed to a user-defined function, CGI functions are often used for system configuration and control. CGI functions may also be invoked through a Server Side Include directive.

AJAX (Asynchronous JavaScript and XML)
AJAX is a term to describe a set of techniques for linking JavaScript functions to a web server to provide responsive web pages that don't require a mouse click from the user in order to update the information that is displayed.

The details of a working AJAX system are somewhat complicated, but the following brief summary may be helpful. A web page that implements AJAX includes JavaScript that is delivered with the web page and which runs on the browser. The JavaScript may include features such as timer-driven periodic loops, text boxes, or mouse-driven controls, so that the page may react to a variety of keyboard or mouse inputs, or simply refresh periodically.

When a screen update requires interaction with the server occurs, the JavaScript uses the XMLHttpRequest object to pass the name of a CGI function and its parameters to the web server. The web server invokes the function and returns results to the web browser. Then the JavaScript logic interprets the results and may display new information. The key benefit here is that the browser no longer sends a request to the server based on a page load or form submission. The request can be sent based on any condition that JavaScript can detect, which includes mouse movement, individual keystrokes, or simply the passage of time. This allows creating web pages that can be used to more directly control or monitor the system that is running the web server.

File System Interface

If file system support is present, then the web server can retrieve pages from the file system in addition to the non-volatile pages that are built with the system image. These files are searched first, so it is also possible to replace a default page that is in ROM with an updated copy in the file system. Web pages that are retrieved from the file system can be created or updated by using removable media, or by using FTP to upload new content over a network connection. A RAM disk can be used to store updated pages, but the pages will be lost if the system loses power.

Access Control

The web server can be configured to restrict access to certain pages using either Basic Authentication or Digest Authentication. When a web browser attempts to retrieve an access controlled page on the web server, the server response will indicate that authentication is required and will list acceptable methods in an HTTP header. The web browser will then present a log in screen to the user, and the user name and password from this form will be sent to the web server for authentication. The web server will return the originally requested page if all is in order. Note that the Basic Authentication does not encrypt the login information, but the Digest Authentication does.

Creating the System

The web pages served by the system can be created using standard web design tools. Once the set of HTML files have been created, the developer can include references to these files in the web server configuration file, and then run an included utility to convert the information into a data file. This data file can then be linked into the ROM section of the system image so that the web pages become part of the system.

The SMX evaluation kits and product shipments that include the web server are supplied with a set of

sample web pages and configuration files that can be customized and extended to become the web page content that ships with the final system. The pages themselves are located in smx\demo\webpage*.htm and may easily be modified to demonstrate the ability to create new web pages. When working with IAR tools, the project files that control the build can automatically invoke the conversion utility, so that the process is entirely accomplished within the IDE. With CodeWarrior tools, the conversion utility is currently invoked by hand from the command line, and then the freshly converted data files can be built into the system image using the usual methods.

The default web pages demonstrate static web pages, HTML form processing via a CGI function, server side includes, and an AJAX control.

smxNS Performance (KB/sec.)

Family	Processor	MHz	Memory	Ethernet	TCP S	TCP R
ARM7	AT91SAM7X256	48	SRAM	on-chip	1315	827
ARM7	LPC2468	72	SRAM	on-chip	966	725
ARM9	AT91SAM9260	210	SDRAM	on-chip	857	532
CF	MCF5282	64	SDRAM	on-chip	1131	845
CF	MCF5329	240	SRAM	on-chip	3048	3276
x86	VIA C3	800	SDRAM	Intel i825xx	6687	6375

S = Send R = Recieve

Notes

- MHz is the clock speed we tested, not necessarily the top speed of the processor.
- SRAM is internal memory on the processor, and SDRAM is external memory.
- VIA C3 may be equivalent to 266MHz Pentium II.
- Benchmarking performed using nuttcp v 5.3.1.

smxNS Sizes

Configuration

Setting	Name	Value
Network Interfaces	NNETS	1
Connections	NCONNS	4
Router Table Entries	NCONFIGS	8
Frame Buffers	NBUFFS	5

Memory Requirements (KB)

Component	ARM Thumb		ARM		ColdFire	
	RAM	ROM	RAM	ROM	RAM	ROM
Core Library	2.0 + 10.0	25.6	2.0 + 10.0	37.1	2.4 + 11.4	48.8
DPI API	0.0	2.2	0.0	3.3	0.1	3.3
Socket API	0.1	4.4	0.1	6.4	0.3	6.3
DHCP c	0.1	3.2	0.1	4.5	0.1	5.0
DHCP s	2.0 + 0.1	5.7	2.0 + 0.1	8.2	2.4 + 0.3	8.9
FTP c	1.0 + 0.6	3.4	1.0 + 0.6	4.5	2.4 + 0.7	7.0
FTP s	2.7 + 0.1	3.2	2.7 + 0.1	4.2	2.7 + 0.3	4.7
HTTP c	1.0 + 1.5	3.2	1.0 + 1.5	3.8	2.4 + 1.5	5.6
IGMP	0.4	2.9	0.4	3.8	0.8	4.8
NAT	0.4	4.0	0.4	8.7	0.5	3.5
POP3 c	tbd + 0.0	1.6	tbd + 0.0	2.3	tbd + 0.1	4.0
PPP	8.0	15.4	8.0	25.1	2.6	22.4
PPPoE c	8.5	18.6	8.5	29.7	3.0	27.3
PPPoE s	8.4	18.8	8.4	29.9	2.9	27.4
SMTP c	tbd + 0.0	2.1	tbd + 0.0	2.8	tbd + 0.1	4.4
SMTP s	tbd + 0.3	1.8	tbd + 0.3	2.5	tbd + 0.4	4.3
SNMP v2	2.7 + 4.0	15.2	2.7 + 4.0	22.4	2.7 + 4.7	18.4
SNMP v3	2.7 + 9.5	25.8	2.7 + 9.5	39.0	2.7 + 10.4	30.2
Telnet Server	0.6 + 0.0	0.7	0.6 + 0.0	0.7	1.2 + 0.0	1.3
Web Server	2.7 + 7.0	12.3	2.7 + 7.0	16.9	3.0 + 8.0	19.3

Notes

- In the RAM columns, the second or only number is the data requirement. The first number is the approximate stack usage for a multitasking system. Otherwise, in a non-multitasking system, ignore those and assume about 3 to 4KB extra stack depth. Some applications, such as the web server, have extra deep stack needs if features such as web form processing are used. tbd indicates we have not yet measured the size.
- These memory requirements are typical for a system that services one active TCP session at a time.
- The Core Library includes support for TCP, UDP, IP, ICMP, ARP, DNS c, and an Ethernet driver.
- Socket API and DHCP c support are commonly used but listed separately.
- Support for IP fragmentation and reassembly is included.
- Support for IP Options Headers is not included.
- PPPoE client and server values include PPP
- For each additional active session, smxNS should be configured with NCONNS increased by 1, NCONFIGS by 1, and NBUFFS by 5. So each active session (client or server) adds about 8KB to the RAM requirement.